



ANEXOS

ÍNDICE DE LOS ANEXOS

| | | |
|--------------------|-------------------------------------|--------|
| Anexo I. | Scripts..... | III |
| Anexo I.I. | Script LoadScene | III |
| Anexo I.II. | Script Timer | IV |
| Anexo I.III. | Script PuertaMovimiento | V |
| Anexo I.IV. | Script GameOver | VI |
| Anexo I.V. | Script ObjectivesCount | VII |
| Anexo I.VI. | Script Selección | VIII |
| Anexo I.VI.I. | Script BotonAutoSeleccion | VIII |
| Anexo I.VI.II. | Script InterruptorCO2Seleccion..... | IX |
| Anexo I.VI.III. | Script PanelAireSeleccion | X |
| Anexo I.VII. | Script RespuestaOk..... | XI |
| Anexo I.VII.I. | Script ACok..... | XI |
| Anexo I.VII.II. | Script Aireok | XII |
| Anexo I.VII.III. | Script Avisotelefonook..... | XIII |
| Anexo I.VII.IV. | Script BotonAutook | XIV |
| Anexo I.VII.V. | Script CartuchosCalok | XV |
| Anexo I.VII.VI. | Script EquipoGasesok | XVI |
| Anexo I.VII.VII. | Script InterruptorCO2ok | XVII |
| Anexo I.VII.VIII. | Script Oxigenook..... | XVIII |
| Anexo I.VII.IX. | Script PanelAireok | XIX |
| Anexo I.VII.X. | Script Puertaok..... | XX |
| Anexo I.VIII. | Script objetivo | XXI |
| Anexo I.VIII.I. | Script AC | XXI |
| Anexo I.VIII.II. | Script BotellasAire | XXIII |
| Anexo I.VIII.III. | Script BotonAuto | XXIV |
| Anexo I.VIII.IV. | Script CartuchosCal | XXV |
| Anexo I.VIII.V. | Script EquipoGases | XXVI |
| Anexo I.VIII.VI. | Script InterruptorCO2..... | XXVII |
| Anexo I.VIII.VII. | Script Oxigeno..... | XXVIII |
| Anexo I.VIII.VIII. | Script PanelAire | XXIX |
| Anexo I.VIII.IX. | Script Puerta | XXX |
| Anexo I.VIII.X. | Script Telefono | XXXI |
| Anexo II. | Cuestionarios de evaluación | XXXII |

Anexo I. Scripts

A continuación, se presentan los *Scripts* creados para la programación de la simulación de la puesta en marcha de la cabina de salvamento.

Anexo I.I. *Script LoadScene*

El *Script* creado para realizar el cambio de escena al clicar una vez con el botón derecho del ratón sobre el cartel “Empezar” es el siguiente:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class LoadScene : MonoBehaviour
7  {
8      public int SceneIndex;
9      void Update()
10     {
11         if (Input.GetMouseButtonDown(0))
12         {
13             SceneLoad(SceneIndex);
14         }
15     }
16     public void SceneLoad(int SceneIndex)
17     {
18         SceneManager.LoadScene(SceneIndex);
19     }
20 }
```

Anexo I.II. *Script Timer*

El *Script* creado para introducir un cronómetro en la ventana de la simulación es el siguiente:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  public class Timer : MonoBehaviour
6  {
7      public Text timerText;
8      private float startTime;
9      public bool finish;
10     void Start()
11     {
12         startTime = Time.time;
13     }
14     void Update()
15     {
16         float t = Time.timeSinceLevelLoad;
17
18         string minutes = ((int) t / 60).ToString();
19         string seconds = (t % 60).ToString("f2");
20
21         timerText.text = minutes + ":" + seconds;
22
23         if (finish)
24         {
25             timerText.text = minutes + ":" + seconds;
26         }
27     }
28     public void EndTime()
29     {
30         finish = true;
31     }
32 }
```


Anexo I.III. Script *PuertaMovimiento*

El *Script* creado para poder abrir y cerrar la puerta de la cabina de salvamento es el siguiente:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class PuertaMovimiento : MonoBehaviour
6  {
7      public float speed;
8      public float angle;
9      public Vector3 direction;
10
11     void Start()
12     {
13         angle = transform.eulerAngles.y;
14     }
15     void Update()
16     {
17         if (Mathf.Round(transform.eulerAngles.y) !=angle)
18         {
19             transform.Rotate(direction * speed);
20         }
21
22         if (Input.GetKeyDown(KeyCode.O))
23         {
24             angle = 120;
25             direction = Vector3.up;
26         }
27
28         if (Input.GetKeyDown(KeyCode.C))
29         {
30             angle = 0;
31             direction = Vector3.down;
32         }
33     }
34 }
```

Anexo I.IV. Script *GameOver*

El *Script* creado para parar la simulación y mostrar el cuadro de texto de puesta en marcha fallida cuando se seleccione una respuesta incorrecta es el siguiente:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  using UnityEngine.Events;
6
7  public class GameOver : MonoBehaviour
8  {
9      public GameObject Canvas;
10     public CharacterController controller;
11     public bool end;
12     void Start()
13     {
14
15     }
16
17     void Update()
18     {
19
20     }
21
22     public void CallGameOver()
23     {
24         end = true;
25         Canvas.GetComponent<Canvas>().enabled = true;
26         controller.enabled = false;
27         Timer script = gameObject.GetComponent<Timer>();
28         Destroy(script);
29     }
30 }
```

Anexo I.V. *Script ObjectivesCount.*

El *Script* creado para que se muestre en pantalla el cuadro de texto de puesta en marcha lograda es el siguiente:

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class ObjectivesCount : MonoBehaviour
6  {
7      public GameObject Canvas;
8      public CharacterController controller;
9
10     public int numObjectives;
11
12     void Start()
13     {
14
15     }
16
17     void Update()
18     {
19         if (numObjectives >= 10)
20         {
21             Canvas.GetComponent<Canvas>().enabled = true;
22             controller.enabled = false;
23             Timer script = gameObject.GetComponent<Timer>();
24             Destroy(script);
25         }
26     }
27 }
```

Anexo I.VI. *Script Selección*

Se ha creado un *Script* para cada elemento del cuadro de mando con el objetivo de que éste se muestre al pasar con el puntero central rojo por encima de cada uno de ellos. El resultado se muestra a continuación:

Anexo I.VI.I. *Script BotonAutoSeleccion*

```

1  using UnityEngine;
2
3  public class BotonAutoC02Seleccion : MonoBehaviour
4  {
5      public GameObject canvas;
6      public Transform cameraTransform;
7      public bool canSeeCanvas = true;
8
9      void Start(){
10
11          canSeeCanvas = true;
12      }
13
14      void Update()
15      {
16          RaycastHit hit;
17          Vector3 rayPos = cameraTransform.position + cameraTransform.forward *0.5f;
18          Debug.DrawRay(rayPos, cameraTransform.forward *1, Color.green, 3f);
19
20          if (Physics.Raycast(rayPos, cameraTransform.forward, out hit, 3f))
21          {
22              if (canSeeCanvas)
23              {
24                  if (hit.transform.tag == "botonAuto")
25                  {
26                      canvas.GetComponent<Canvas>().enabled=true;
27                  }
28                  else
29                  {
30                      canvas.GetComponent<Canvas>().enabled=false;
31                  }
32              }
33          }
34          else
35          {
36              canvas.GetComponent<Canvas>().enabled=false;
37          }
38      }
39  }

```

Anexo I.VI.II. *Script InterruptorCO2Seleccion*

```
1  using UnityEngine;
2
3  public class InterruptorCO2Seleccion : MonoBehaviour
4  {
5      public GameObject canvas;
6      public Transform cameraTransform;
7      public bool canSeeCanvas = true;
8
9      void Start(){
10
11          canSeeCanvas = true;
12      }
13
14      void Update()
15      {
16          RaycastHit hit;
17          Vector3 rayPos = cameraTransform.position + cameraTransform.forward *0.5f;
18          Debug.DrawRay(rayPos, cameraTransform.forward *1, Color.green, 3f);
19
20          if (Physics.Raycast(rayPos, cameraTransform.forward, out hit, 3f))
21          {
22              if (canSeeCanvas)
23              {
24                  if (hit.transform.tag == "interruptorCO2")
25                  {
26                      canvas.GetComponent<Canvas>().enabled=true;
27                  }
28                  else
29                  {
30                      canvas.GetComponent<Canvas>().enabled=false;
31                  }
32              }
33          }
34          else
35          {
36              canvas.GetComponent<Canvas>().enabled=false;
37          }
38      }
39  }
```

Anexo I.VI.III. *Script PanelAireSeleccion*

```
1  using UnityEngine;
2
3  public class PanelAireSeleccion : MonoBehaviour
4  {
5      public GameObject canvas;
6      public Transform cameraTransform;
7      public bool canSeeCanvas = true;
8
9      void Start()
10
11          canSeeCanvas = true;
12  }
13
14  void Update()
15  {
16      RaycastHit hit;
17      Vector3 rayPos = cameraTransform.position + cameraTransform.forward *0.5f;
18      Debug.DrawRay(rayPos, cameraTransform.forward *1, Color.green, 3f);
19
20      if (Physics.Raycast(rayPos, cameraTransform.forward, out hit, 3f))
21      {
22          if (canSeeCanvas)
23          {
24              if (hit.transform.tag == "panelAire")
25              {
26                  canvas.GetComponent<Canvas>().enabled=true;
27              }
28              else
29              {
30                  canvas.GetComponent<Canvas>().enabled=false;
31              }
32          }
33      }
34      else
35      {
36          canvas.GetComponent<Canvas>().enabled=false;
37      }
38  }
39  }
```

Anexo I.VII. *Script RespuestaOk*

Los *Scripts* creados para mostrar los cuadros de texto de respuesta y hacerlos desaparecer al cabo de 2 segundos son los que se detallan a continuación. Se crea uno para cada paso de la puesta en marcha de la cabina de salvamento.

Anexo I.VII.I. *Script ACok*

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class ACok : MonoBehaviour
7  {
8      public GameObject ACokText;
9      public GameObject Check10;
10     public bool startTime;
11     public float temps;
12     public GameObject estrella;
13
14     void Start()
15     {
16
17     }
18
19     void Update()
20     {
21         if (startTime)
22         {
23             temps += Time.deltaTime;
24         }
25
26         if (temps > 2)
27         {
28             ACokText.GetComponent<Canvas>().enabled = false;
29         }
30     }
31
32     public void CallACok()
33     {
34         startTime = true;
35         ACokText.GetComponent<Canvas>().enabled = true;
36         estrella.SetActive(false);
37         Check10.GetComponent<Image>().enabled = true;
38     }
39 }
```

Anexo I.VII.II. *Script Aireok*

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class Aireok : MonoBehaviour
7  {
8      public GameObject AireokText;
9      public GameObject Check2;
10     public bool startTime;
11     public float temps;
12     public GameObject estrella;
13
14     void Start()
15     {
16
17     }
18
19     void Update()
20     {
21         if (startTime)
22         {
23             temps += Time.deltaTime;
24         }
25
26         if (temps > 2)
27         {
28             AireokText.GetComponent<Canvas>().enabled = false;
29         }
30     }
31
32     public void CallAireok()
33     {
34         startTime = true;
35         AireokText.GetComponent<Canvas>().enabled = true;
36         estrella.SetActive(false);
37         Check2.GetComponent<Image>().enabled = true;
38     }
39 }
```


Anexo I.VII.III. *Script Avisotelefonook*

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class Avisotelefonook : MonoBehaviour
7  {
8      public GameObject AvisotelefonookText;
9      public GameObject Check9;
10     public bool startTime;
11     public float temps;
12     public GameObject estrella;
13
14     void Start()
15     {
16
17     }
18
19     void Update()
20     {
21         if (startTime)
22         {
23             temps += Time.deltaTime;
24         }
25
26         if (temps > 2)
27         {
28             AvisookText.GetComponent<Canvas>().enabled = false;
29         }
30     }
31
32     public void CallAvisotelefonook()
33     {
34         startTime = true;
35         AvisookText.GetComponent<Canvas>().enabled = true;
36         estrella.SetActive(false);
37         Check9.GetComponent<Image>().enabled = true;
38     }
39 }
```

Anexo I.VII.IV. *Script BotonAutook*

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class BotonAutook : MonoBehaviour
7  {
8      public GameObject BotonAutookText;
9      public GameObject Check4;
10     public bool startTime;
11     public float temps;
12     public GameObject estrella;
13
14     void Start()
15     {
16
17     }
18
19     void Update()
20     {
21         if (startTime)
22         {
23             temps += Time.deltaTime;
24         }
25
26         if (temps > 2)
27         {
28             BotonAutookText.GetComponent<Canvas>().enabled = false;
29         }
30     }
31
32     public void CallBotonAutook()
33     {
34         startTime = true;
35         BotonAutookText.GetComponent<Canvas>().enabled = true;
36         estrella.SetActive(false);
37         Check4.GetComponent<Image>().enabled = true;
38     }
39 }
```

Anexo I.VII.V. *Script CartuchosCalok*

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class CartuchosCalok : MonoBehaviour
7  {
8      public GameObject CartuchosCalokText;
9      public GameObject Check6;
10     public bool startTime;
11     public float temps;
12     public GameObject estrella;
13
14     void Start()
15     {
16
17     }
18
19     void Update()
20     {
21         if (startTime)
22         {
23             temps += Time.deltaTime;
24         }
25
26         if (temps > 2)
27         {
28             CartuchosCalokText.GetComponent<Canvas>().enabled = false;
29         }
30     }
31
32     public void CallCartuchosCalok()
33     {
34         startTime = true;
35         CartuchosCalokText.GetComponent<Canvas>().enabled = true;
36         estrella.SetActive(false);
37         Check6.GetComponent<Image>().enabled = true;
38     }
39 }
```

Anexo I.VII.VI. *Script EquipoGasesok*

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class EquipoGasesok : MonoBehaviour
7  {
8      public GameObject EquipoGasesokText;
9      public GameObject Check8;
10     public bool startTime;
11     public float temps;
12     public GameObject estrella;
13
14     void Start()
15     {
16
17     }
18
19     void Update()
20     {
21         if (startTime)
22         {
23             temps += Time.deltaTime;
24         }
25
26         if (temps > 2)
27         {
28             EquipoGasesokText.GetComponent<Canvas>().enabled = false;
29         }
30     }
31
32     public void CallEquipoGasesok()
33     {
34         startTime = true;
35         EquipoGasesokText.GetComponent<Canvas>().enabled = true;
36         estrella.SetActive(false);
37         Check8.GetComponent<Image>().enabled = true;
38     }
39 }
```

Anexo I.VII.VII. *Script InterruptorCO2ok*

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class InterruptorCO2ok : MonoBehaviour
7  {
8      public GameObject InterruptorCO2okText;
9      public GameObject Check5;
10     public bool startTime;
11     public float temps;
12     public GameObject estrella;
13
14     void Start()
15     {
16
17     }
18
19     void Update()
20     {
21         if (startTime)
22         {
23             temps += Time.deltaTime;
24         }
25
26         if (temps > 2)
27         {
28             InterruptorCO2okText.GetComponent<Canvas>().enabled = false;
29         }
30     }
31
32     public void CallInterruptorCO2ok()
33     {
34         startTime = true;
35         InterruptorCO2okText.GetComponent<Canvas>().enabled = true;
36         estrella.SetActive(false);
37         Check5.GetComponent<Image>().enabled = true;
38     }
39 }
```

Anexo I.VII.VIII. *Script Oxigenook*

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class Oxigenook : MonoBehaviour
7  {
8      public GameObject OxigenookText;
9      public GameObject Check7;
10     public bool startTime;
11     public float temps;
12     public GameObject estrella;
13
14     void Start()
15     {
16
17     }
18
19     void Update()
20     {
21         if (startTime)
22         {
23             temps += Time.deltaTime;
24         }
25
26         if (temps > 2)
27         {
28             OxigenookText.GetComponent<Canvas>().enabled = false;
29         }
30     }
31
32     public void CallOxigenook()
33     {
34         startTime = true;
35         OxigenookText.GetComponent<Canvas>().enabled = true;
36         estrella.SetActive(false);
37         Check7.GetComponent<Image>().enabled = true;
38     }
39 }
```

Anexo I.VII.IX. *Script PanelAireok*

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class PanelAireok : MonoBehaviour
7  {
8      public GameObject PanelAireokText;
9      public GameObject Check3;
10     public bool startTime;
11     public float temps;
12     public GameObject estrella;
13
14     void Start()
15     {
16
17     }
18
19     void Update()
20     {
21         if (startTime)
22         {
23             temps += Time.deltaTime;
24         }
25
26         if (temps > 2)
27         {
28             PanelAireokText.GetComponent<Canvas>().enabled = false;
29         }
30     }
31
32     public void CallPanelAireok()
33     {
34         startTime = true;
35         PanelAireokText.GetComponent<Canvas>().enabled = true;
36         estrella.SetActive(false);
37         Check3.GetComponent<Image>().enabled = true;
38     }
39 }
```

Anexo I.VII.X. *Script Puertaok*

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class Puertaok : MonoBehaviour
7  {
8      public GameObject PPuertaokText;
9      public GameObject Check1;
10     public bool startTime;
11     public float temps;
12     public GameObject estrella;
13
14     void Start()
15     {
16
17     }
18
19     void Update()
20     {
21         if (startTime)
22         {
23             temps += Time.deltaTime;
24         }
25
26         if (temps > 2)
27         {
28             PuertaokText.GetComponent<Canvas>().enabled = false;
29         }
30     }
31
32     public void CallPuertaok()
33     {
34         startTime = true;
35         PuertaokText.GetComponent<Canvas>().enabled = true;
36         estrella.SetActive(false);
37         Check1.GetComponent<Image>().enabled = true;
38     }
39 }
```


Anexo I.VIII. Script objetivo

Los *Scripts* creados para mostrar los cuadros de texto con las preguntas cuando al pasar el puntero central rojo por encima de los diferentes objetos se presentan a continuación. Se crea uno para cada paso de la puesta en marcha de la cabina de salvamento.

Anexo I.VIII.I. Script AC

```

1  using UnityEngine;
2
3  public class AC : MonoBehaviour
4  {
5      public GameObject canvas;
6      public Transform cameraTransform;
7      public bool canSeeCanvas = true;
8
9      void Start()
10     {
11         canSeeCanvas = true;
12     }
13     void Update()
14     {
15         RaycastHit hit;
16         Vector3 rayPos = cameraTransform.position + cameraTransform.forward * 0.5f;
17         Debug.DrawRay(rayPos, cameraTransform.forward * 1, Color.green, 3f);
18
19         if (Physics.Raycast(rayPos, cameraTransform.forward, out hit, 3f))
20         {
21             if (canSeeCanvas)
22             {
23                 if (hit.transform.tag == "mandoAC")
24                 {
25                     canvas.GetComponent<Canvas>().enabled=true;
26
27                     if (Input.GetKeyDown(KeyCode.M))
28                     {
29                         canvas.SetActive(false);
30                         canSeeCanvas = false;
31                         GameObject final = gameObject.GetComponent<GameOver>();
32                         final.CallGameOver();
33                     }
34                     if (Input.GetKeyDown(KeyCode.N))
35                     {
36                         canvas.SetActive(false);
37                         canSeeCanvas = false;
38                         ACok final = gameObject.GetComponent<ACok>();
39                         ObjectivesCount script =
40                         = gameObject.GetComponent<ObjectivesCount>();
41                         script.numObjectives += 1;
42                         final.CallACok();
43                     }
44                     if (Input.GetKeyDown(KeyCode.B))
45                     {
46                         canvas.SetActive(false);
47                         canSeeCanvas = false;
48                         GameObject final = gameObject.GetComponent<GameOver>();
49                         final.CallGameOver();
50                     }
51                 }
52             }
53             else
54             {
55                 canvas.GetComponent<Canvas>().enabled=false;
56             }
57         }
58     }
59 }

```

```
57         else
58         {
59             canvas.GetComponent<Canvas>().enabled=false;
60         }
61     }
62 }
```

Anexo I.VIII.II. *Script BotellasAire*

```
1  using UnityEngine;
2
3  public class BotellasAire: MonoBehaviour
4  {
5      public GameObject canvas;
6      public Transform cameraTransform;
7      public bool canSeeCanvas = true;
8
9      void Start()
10     {
11         canSeeCanvas = true;
12     }
13     void Update()
14     {
15         RaycastHit hit;
16         Vector3 rayPos = cameraTransform.position + cameraTransform.forward *0.5f;
17         Debug.DrawRay(rayPos, cameraTransform.forward *1, Color.green, 3f);
18
19         if (Physics.Raycast(rayPos, cameraTransform.forward, out hit, 3f))
20         {
21             if (canSeeCanvas)
22             {
23                 if (hit.transform.tag == "aire")
24                 {
25                     canvas.GetComponent<Canvas>().enabled=true;
26
27                     if (Input.GetKeyDown(KeyCode.M))
28                     {
29                         canvas.SetActive(false);
30                         canSeeCanvas = false;
31                         GameObject final = gameObject.GetComponent<GameOver>();
32                         final.CallGameOver();
33                     }
34                     if (Input.GetKeyDown(KeyCode.N))
35                     {
36                         canvas.SetActive(false);
37                         canSeeCanvas = false;
38                         Aireok final = gameObject.GetComponent<Aireok>();
39                         ObjectivesCount script =
40                         = gameObject.GetComponent<ObjectivesCount>();
41                         script.numObjectives += 1;
42                         final.CallAireok();
43                     }
44                     if (Input.GetKeyDown(KeyCode.B))
45                     {
46                         canvas.SetActive(false);
47                         canSeeCanvas = false;
48                         GameObject final = gameObject.GetComponent<GameOver>();
49                         final.CallGameOver();
50                     }
51                 }
52             }
53             else
54             {
55                 canvas.GetComponent<Canvas>().enabled=false;
56             }
57         }
58         else
59         {
60             canvas.GetComponent<Canvas>().enabled=false;
61         }
62     }
```

Anexo I.VIII.III. *Script BotonAuto*

```

1  using UnityEngine;
2
3  public class BotonAuto : MonoBehaviour
4  {
5      public GameObject canvas;
6      public Transform cameraTransform;
7      public bool canSeeCanvas = true;
8
9      void Start()
10     {
11         canSeeCanvas = true;
12     }
13     void Update()
14     {
15         RaycastHit hit;
16         Vector3 rayPos = cameraTransform.position + cameraTransform.forward *0.5f;
17         Debug.DrawRay(rayPos, cameraTransform.forward *1, Color.green, 3f);
18
19         if (Physics.Raycast(rayPos, cameraTransform.forward, out hit, 3f))
20         {
21             if (canSeeCanvas)
22             {
23                 if (hit.transform.tag == "botonauto")
24                 {
25                     canvas.GetComponent<Canvas>().enabled=true;
26
27                     if (Input.GetKeyDown(KeyCode.M))
28                     {
29                         canvas.SetActive(false);
30                         canSeeCanvas = false;
31                         BotonAutook final = gameObject.GetComponent<BotonAutook>();
32                         ObjectivesCount script =
33                             = gameObject.GetComponent<ObjectivesCount>();
34                         script.numObjectives += 1;
35                         final.CallBotonAutook();
36                     }
37                     if (Input.GetKeyDown(KeyCode.N))
38                     {
39                         canvas.SetActive(false);
40                         canSeeCanvas = false;
41                         GameOver final = gameObject.GetComponent<GameOver>();
42                         final.CallGameOver();
43                     }
44                     if (Input.GetKeyDown(KeyCode.B))
45                     {
46                         canvas.SetActive(false);
47                         canSeeCanvas = false;
48                         GameOver final = gameObject.GetComponent<GameOver>();
49                         final.CallGameOver();
50                     }
51                 }
52             }
53             else
54             {
55                 canvas.GetComponent<Canvas>().enabled=false;
56             }
57         }
58         else
59         {
60             canvas.GetComponent<Canvas>().enabled=false;
61         }
62     }
63 }

```

Anexo I.VIII.IV. *Script CartuchosCal*

```

1  using UnityEngine;
2
3  public class CartuchosCal : MonoBehaviour
4  {
5      public Transform cameraTransform;
6      public GameObject cartuchosMaquina;
7      public GameObject cartuchosSuelo;
8      public GameObject canvas;
9      public bool canSeeCanvas = true;
10
11     void Start()
12     {
13         canSeeCanvas = true;
14     }
15     void Update()
16     {
17         RaycastHit hit;
18         Vector3 rayPos = cameraTransform.position + cameraTransform.forward * 0.5f;
19
20         if (Physics.Raycast(rayPos, cameraTransform.forward, out hit, 3f))
21         {
22             if (canSeeCanvas)
23             {
24                 if (hit.transform.tag == "cartuchos")
25                 {
26                     canvas.GetComponent<Canvas>().enabled=true;
27
28                     if (Input.GetKeyDown(KeyCode.M))
29                     {
30                         canvas.SetActive(false);
31                         canSeeCanvas = false;
32                         GameObject final = gameObject.GetComponent<GameOver>();
33                         final.CallGameOver();
34                     }
35                     if (Input.GetKeyDown(KeyCode.N))
36                     {
37                         canvas.SetActive(false);
38                         canSeeCanvas = false;
39                         GameObject final = gameObject.GetComponent<GameOver>();
40                         final.CallGameOver();
41                     }
42                     if (Input.GetKeyDown(KeyCode.B))
43                     {
44                         cartuchosSuelo.SetActive(false);
45                         cartuchosMaquina.SetActive(true);
46                         canvas.SetActive(false);
47                         canSeeCanvas = false;
48                         CartuchosCalok final =
49                             = gameObject.GetComponent<CartuchosCalok>();
50                         ObjectivesCount script =
51                             = gameObject.GetComponent<ObjectivesCount>();
52                         script.numObjectives += 1;
53                         final.CallCartuchosCalok();
54                     }
55                 }
56             }
57             else
58             {
59                 canvas.GetComponent<Canvas>().enabled=false;
60             }
61         }
62         else
63         {
64             canvas.GetComponent<Canvas>().enabled=false;
65         }
66     }
67 }

```

Anexo I.VIII.V. *Script EquipoGases*

```

1  using UnityEngine;
2
3  public class EquipoGases : MonoBehaviour
4  {
5      public GameObject canvas;
6      public Transform cameraTransform;
7      public bool canSeeCanvas = true;
8
9      void Start()
10     {
11         canSeeCanvas = true;
12     }
13     void Update()
14     {
15         RaycastHit hit;
16         Vector3 rayPos = cameraTransform.position + cameraTransform.forward *0.5f;
17         Debug.DrawRay(rayPos, cameraTransform.forward *1, Color.green, 3f);
18
19         if (Physics.Raycast(rayPos, cameraTransform.forward, out hit, 3f))
20         {
21             if (canSeeCanvas)
22             {
23                 if (hit.transform.tag == "medidorgases")
24                 {
25                     canvas.GetComponent<Canvas>().enabled=true;
26
27                     if (Input.GetKeyDown(KeyCode.M))
28                     {
29                         canvas.SetActive(false);
30                         canSeeCanvas = false;
31                         EquipoGasesok final =
32                         = gameObject.GetComponent<EquipoGasesok>();
33                         ObjectivesCount script =
34                         = gameObject.GetComponent<ObjectivesCount>();
35                         script.numObjectives += 1;
36                         final.CallEquipoGasesok();
37                     }
38                     if (Input.GetKeyDown(KeyCode.N))
39                     {
40                         canvas.SetActive(false);
41                         canSeeCanvas = false;
42                         GameOver final = gameObject.GetComponent<GameOver>();
43                         final.CallGameOver();
44                     }
45                     if (Input.GetKeyDown(KeyCode.B))
46                     {
47                         canvas.SetActive(false);
48                         canSeeCanvas = false;
49                         GameOver final = gameObject.GetComponent<GameOver>();
50                         final.CallGameOver();
51                     }
52                 }
53             }
54             else
55             {
56                 canvas.GetComponent<Canvas>().enabled=false;
57             }
58         }
59         else
60         {
61             canvas.GetComponent<Canvas>().enabled=false;
62         }
63     }
64 }

```

Anexo I.VIII.VI. *Script InterruptorCO2*

```

1  using UnityEngine;
2
3  public class InterruptorCO2 : MonoBehaviour
4  {
5      public GameObject canvas;
6      public Transform cameraTransform;
7      public bool canSeeCanvas = true;
8
9      void Start()
10     {
11         canSeeCanvas = true;
12     }
13     void Update()
14     {
15         RaycastHit hit;
16         Vector3 rayPos = cameraTransform.position + cameraTransform.forward *0.5f;
17         Debug.DrawRay(rayPos, cameraTransform.forward *1, Color.green, 3f);
18
19         if (Physics.Raycast(rayPos, cameraTransform.forward, out hit, 3f))
20         {
21             if (canSeeCanvas)
22             {
23                 if (hit.transform.tag == "interruptorCO2")
24                 {
25                     canvas.GetComponent<Canvas>().enabled=true;
26
27                     if (Input.GetKeyDown(KeyCode.M))
28                     {
29                         canvas.SetActive(false);
30                         canSeeCanvas = false;
31                         GameObject final = gameObject.GetComponent<GameOver>();
32                         final.CallGameOver();
33                     }
34                     if (Input.GetKeyDown(KeyCode.N))
35                     {
36                         canvas.SetActive(false);
37                         canSeeCanvas = false;
38                         InterruptorCO2ok final =
39                         = gameObject.GetComponent<InterruptorCO2ok>();
40                         ObjectivesCount script =
41                         = gameObject.GetComponent<ObjectivesCount>();
42                         script.numObjectives += 1;
43                         final.CallInterruptorCO2ok();
44                     }
45                     if (Input.GetKeyDown(KeyCode.B))
46                     {
47                         canvas.SetActive(false);
48                         canSeeCanvas = false;
49                         GameObject final = gameObject.GetComponent<GameOver>();
50                         final.CallGameOver();
51                     }
52                 }
53             }
54             else
55             {
56                 canvas.GetComponent<Canvas>().enabled=false;
57             }
58         }
59         else
60         {
61             canvas.GetComponent<Canvas>().enabled=false;
62         }
63     }
64 }

```

Anexo I.VIII.VII. *Script Oxigeno*

```

1  using UnityEngine;
2
3  public class Oxigeno : MonoBehaviour
4  {
5      public GameObject canvas;
6      public Transform cameraTransform;
7      public bool canSeeCanvas = true;
8
9      void Start()
10     {
11         canSeeCanvas = true;
12     }
13     void Update()
14     {
15         RaycastHit hit;
16         Vector3 rayPos = cameraTransform.position + cameraTransform.forward *0.5f;
17         Debug.DrawRay(rayPos, cameraTransform.forward *1, Color.green, 3f);
18
19         if (Physics.Raycast(rayPos, cameraTransform.forward, out hit, 3f))
20         {
21             if (canSeeCanvas)
22             {
23                 if (hit.transform.tag == "oxigeno")
24                 {
25                     canvas.GetComponent<Canvas>().enabled=true;
26
27                     if (Input.GetKeyDown(KeyCode.M))
28                     {
29                         canvas.SetActive(false);
30                         canSeeCanvas = false;
31                         GameObject final = gameObject.GetComponent<GameOver>();
32                         final.CallGameOver();
33                     }
34                     if (Input.GetKeyDown(KeyCode.N))
35                     {
36                         canvas.SetActive(false);
37                         canSeeCanvas = false;
38                         GameObject final = gameObject.GetComponent<GameOver>();
39                         final.CallGameOver();
40                     }
41                     if (Input.GetKeyDown(KeyCode.B))
42                     {
43                         canvas.SetActive(false);
44                         canSeeCanvas = false;
45                         Oxigenook final = gameObject.GetComponent<Oxigenook>();
46                         ObjectivesCount script =
47                         = gameObject.GetComponent<ObjectivesCount>();
48                         script.numObjectives += 1;
49                         final.CallOxigenook();
50                     }
51                 }
52             }
53             else
54             {
55                 canvas.GetComponent<Canvas>().enabled=false;
56             }
57         }
58         else
59         {
60             canvas.GetComponent<Canvas>().enabled=false;
61         }
62     }

```


Anexo I.VIII.VIII. *Script PanelAire*

```

1  using UnityEngine;
2
3  public class PanelAire : MonoBehaviour
4  {
5      public GameObject canvas;
6      public Transform cameraTransform;
7      public bool canSeeCanvas = true;
8
9      void Start()
10     {
11         canSeeCanvas = true;
12     }
13     void Update()
14     {
15         RaycastHit hit;
16         Vector3 rayPos = cameraTransform.position + cameraTransform.forward *0.5f;
17         Debug.DrawRay(rayPos, cameraTransform.forward *1, Color.green, 3f);
18
19         if (Physics.Raycast(rayPos, cameraTransform.forward, out hit, 3f))
20         {
21             if (canSeeCanvas)
22             {
23                 if (hit.transform.tag == "panelAire")
24                 {
25                     canvas.GetComponent<Canvas>().enabled=true;
26
27                     if (Input.GetKeyDown(KeyCode.M))
28                     {
29                         canvas.SetActive(false);
30                         canSeeCanvas = false;
31                         PanelAireok final =
32                         gameObject.GetComponent<PanelAireok>();
33                         ObjectivesCount script =
34                         = gameObject.GetComponent<ObjectivesCount>();
35                         script.numObjectives += 1;
36                         final.CallPanelAireok();
37                     }
38                     if (Input.GetKeyDown(KeyCode.N))
39                     {
40                         canvas.SetActive(false);
41                         canSeeCanvas = false;
42                         GameOver final = gameObject.GetComponent<GameOver>();
43                         final.CallGameOver();
44                     }
45                     if (Input.GetKeyDown(KeyCode.B))
46                     {
47                         canvas.SetActive(false);
48                         canSeeCanvas = false;
49                         GameOver final = gameObject.GetComponent<GameOver>();
50                         final.CallGameOver();
51                     }
52                 }
53             }
54             else
55             {
56                 canvas.GetComponent<Canvas>().enabled=false;
57             }
58         }
59         else
60         {
61             canvas.GetComponent<Canvas>().enabled=false;
62         }
63     }
64 }

```

Anexo I.VIII.IX. *Script Puerta*

```

1  using UnityEngine;
2
3  public class Puerta : MonoBehaviour
4  {
5      public GameObject Text;
6      public Transform cameraTransform;
7      public bool canSeeText = true;
8
9      void Start()
10     {
11         canSeeText = true;
12     }
13     void Update()
14     {
15         RaycastHit hit;
16         Vector3 rayPos = cameraTransform.position + cameraTransform.forward *0.5f;
17         Debug.DrawRay(rayPos, cameraTransform.forward *1, Color.green, 5f);
18
19         if (Physics.Raycast(rayPos, cameraTransform.forward, out hit, 5f))
20         {
21             if (canSeeText)
22             {
23                 if (hit.transform.tag == "puerta")
24                 {
25                     Text.GetComponent<Text>().enabled=true;
26
27                     if (Input.GetKeyDown(KeyCode.C))
28                     {
29                         Puertaok final = gameObject.GetComponent<Puertaok>();
30                         ObjectivesCount script
31                         = gameObject.GetComponent<ObjectivesCount>();
32                         script.numObjectives += 1;
33                         final.CallPuertaok();
34                     }
35                 }
36             }
37             else
38             {
39                 Text.GetComponent<Text>().enabled=false;
40             }
41         }
42         else
43         {
44             Text.GetComponent<Text>().enabled=false;
45         }
46     }
47 }
```

Anexo I.VIII.X. Script Telefono

```

1  using UnityEngine;
2
3  public class Telefono : MonoBehaviour
4  {
5      public GameObject canvas;
6      public Transform cameraTransform;
7      public bool canSeeCanvas = true;
8
9      void Start()
10     {
11         canSeeCanvas = true;
12     }
13     void Update()
14     {
15         RaycastHit hit;
16         Vector3 rayPos = cameraTransform.position + cameraTransform.forward *0.5f;
17         Debug.DrawRay(rayPos, cameraTransform.forward *1, Color.green, 3f);
18
19         if (Physics.Raycast(rayPos, cameraTransform.forward, out hit, 3f))
20         {
21             if (canSeeCanvas)
22             {
23                 if (hit.transform.tag == "Telefono")
24                 {
25                     canvas.GetComponent<Canvas>().enabled=true;
26
27                     if (Input.GetKeyDown(KeyCode.M))
28                     {
29                         canvas.SetActive(false);
30                         canSeeCanvas = false;
31                         GameObject final = gameObject.GetComponent<GameOver>();
32                         final.CallGameOver();
33                     }
34                     if (Input.GetKeyDown(KeyCode.N))
35                     {
36                         canvas.SetActive(false);
37                         canSeeCanvas = false;
38                         GameObject final = gameObject.GetComponent<GameOver>();
39                         final.CallGameOver();
40                     }
41                     if (Input.GetKeyDown(KeyCode.B))
42                     {
43                         canvas.SetActive(false);
44                         canSeeCanvas = false;
45                         Avisotelefonook final =
46                         gameObject.GetComponent<Avisotelefonook>();
47                         ObjectivesCount script =
48                         gameObject.GetComponent<ObjectivesCount>();
49                         script.numObjectives += 1;
50                         final.CallAvisotelefonook();
51                     }
52                 }
53             }
54             else
55             {
56                 canvas.GetComponent<Canvas>().enabled=false;
57             }
58         }
59         else
60         {
61             canvas.GetComponent<Canvas>().enabled=false;
62         }
63     }
64 }

```



Anexo II. Cuestionarios de evaluación

***Cuestionario relativo al procedimiento de puesta en marcha de las cabinas de salvamento
ubicadas en las minas de potasa de Cataluña***

Fecha:[illegible]

| Valore los siguientes parámetros (1-muy mal/muy poco; 2- mal/poco; 3-regular; 4-bien/mucho; 5-muy bien/muchísimo) | | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| ¿Cree que la formación le ha ayudado a adquirir conceptos que desconocía o no recordaba? | | | | | |
| ¿Cree que la formación le ha ayudado a adquirir nuevas habilidades? | | | | | |
| ¿Cree que la formación propicia la reflexión? | | | | | |
| ¿Considera que la formación es interesante y fomenta la adquisición de nuevos conocimientos? | | | | | |
| ¿Considera que el soporte digital es interesante y ayuda a adquirir nuevos conocimientos? | | | | | |
| Comentarios: | | | | | |

***Cuestionario relativo al procedimiento de puesta en marcha de las cabinas de salvamento
ubicadas en las minas de potasa de Cataluña***

Fecha:[illegible]

| Valore los siguientes parámetros (1-muy mal/muy poco; 2- mal/poco; 3-regular; 4-bien/mucho; 5-muy bien/muchísimo) | | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| ¿Cree que la formación le ha ayudado a adquirir conceptos que desconocía o no recordaba? | | | | | |
| ¿Cree que la formación le ha ayudado a adquirir nuevas habilidades? | | | | | |
| ¿Cree que la formación propicia la reflexión? | | | | | |
| ¿Considera que la formación es interesante y fomenta la adquisición de nuevos conocimientos? | | | | | |
| ¿Considera que el soporte digital es interesante y ayuda a adquirir nuevos conocimientos? | | | | | |
| Comentarios: | | | | | |

CUESTIONARIO DE LA PARTE TEÓRICA

Cuestionario relativo al procedimiento de puesta en marcha de las cabinas de salvamento ubicadas en las minas de potasa de Cataluña

Nombre: **SERGI CAROL** Edad: **24** Sexo: **M** Fecha: **06/09/19**

Escriba, en orden sucesivo de ejecución, todos los pasos que conozca para poner en marcha la cabina de salvamento

Cerrar Puerta

Abrir botellas aire

Comprovar Posición de válvulas

Activar regulador automática presión

Poner cal en máquina

~~Poner~~ Activar máquina filtradora CO2

Activar aparato medidor Oxígeno - CO2

Llamar a salvamento

Encender aire acondicionado

Valore los siguientes parámetros (1-muy mal/muy poco; 2- mal/poco; 3-regular; 4-bien/mucho; 5 muy bien/muchísimo)

| | 1 | 2 | 3 | 4 | 5 |
|--|---|---|---|---|---|
| ¿Cree que la formación le ha ayudado a adquirir conceptos que desconocía o no recordaba? | | | | X | |
| ¿Cree que la formación le ha ayudado a adquirir nuevas habilidades? | | X | | | |
| ¿Cree que la formación propicia la reflexión? | X | | | | |
| ¿Considera que la formación es interesante y fomenta la adquisición de nuevos conocimientos? | | X | | | |
| ¿Considera que el soporte digital es interesante y ayuda a adquirir nuevos conocimientos? | | X | | | |

Comentarios:

TIEMPO: **4:10.72**

SIMULACIÓN OK.

CUESTIONARIO DE LA PARTE PRÁCTICA

Cuestionario relativo al procedimiento de puesta en marcha de las cabinas de salvamento ubicadas en las minas de potasa de Cataluña

Nombre: **SERGI CAROL** Edad: **29** Sexo: **M** Fecha: **06/09/19**

Escriba, en orden sucesivo de ejecución, todos los pasos que conozca para poner en marcha la cabina de salvamento

Cerrar Puerta de la cabina

Abrir valvulas botellas

Poner las llaves en la posición adecuada

Activar regulador automática presión cabina

Activar interruptor CO2

Poner cartuchos de cal en la máquina

Abrir valvulas Oxígeno de las botellas

Activar el medidor de Oxígeno y CO2

Llamar al centro de salvamento

Encender aire acondicionado

Valore los siguientes parámetros (1-muy mal/muy poco; 2- mal/poco; 3-regular; 4-bien/mucho; 5-muy bien/muchísimo)

| | 1 | 2 | 3 | 4 | 5 |
|--|---|---|---|---|---|
| ¿Cree que la formación le ha ayudado a adquirir conceptos que desconocía o no recordaba? | | | | X | |
| ¿Cree que la formación le ha ayudado a adquirir nuevas habilidades? | | | | X | |
| ¿Cree que la formación propicia la reflexión? | | | X | | |
| ¿Considera que la formación es interesante y fomenta la adquisición de nuevos conocimientos? | | | | | X |
| ¿Considera que el soporte digital es interesante y ayuda a adquirir nuevos conocimientos? | | | | | X |

Comentarios:

CUESTIONARIO DE LA PARTE TEÓRICA

Cuestionario relativo al procedimiento de puesta en marcha de las cabinas de salvamento ubicadas en las minas de potasa de Cataluña

Nombre: Eric De Pablo

Edad: 25

Sexo: A

Fecha: 08/09/19

Escriba, en orden sucesivo de ejecución, todos los pasos que conozca para poner en marcha la cabina de salvamento

1. Entrar y cerrar la puerta
2. Abrir botellas de aire
3. Encender funcionamiento automático
4. Abrir oxígeno
5. Poner cartuchos
6. Encender motor absorbedor
7. Comprobar gases
8. Llamar al centro de control
9. Poner el aire acondicionado
- 10.

Valore los siguientes parámetros (1-muy mal/muy poco; 2- mal/poco; 3-regular; 4-bien/mucho; 5 muy bien/muchísimo)

| | 1 | 2 | 3 | 4 | 5 |
|--|---|---|---|---|---|
| ¿Cree que la formación le ha ayudado a adquirir conceptos que desconocía o no recordaba? | | | X | | |
| ¿Cree que la formación le ha ayudado a adquirir nuevas habilidades? | | | X | | |
| ¿Cree que la formación propicia la reflexión? | X | | | | |
| ¿Considera que la formación es interesante y fomenta la adquisición de nuevos conocimientos? | | | | X | |
| ¿Considera que el soporte digital es interesante y ayuda a adquirir nuevos conocimientos? | | | | X | |

Comentarios:

CUESTIONARIO DE LA PARTE PRÁCTICA

Cuestionario relativo al procedimiento de puesta en marcha de las cabinas de salvamento ubicadas en las minas de potasa de Cataluña

Nombre: Eric De Pablo

Edad: 25

Sexo: M

Fecha: 08/09/19

Escriba, en orden sucesivo de ejecución, todos los pasos que conozca para poner en marcha la cabina de salvamento

1. Entrar y cerrar la puerta.
2. Abrir botellas aire respirable
3. Comprobar las válvulas del panel
4. Poner el autofuncionamiento.
5. Encender el motor del absorbedor
6. Abrir el oxígeno
7. Poner los cartuchos (6 cartuchos)
8. Encender aparato medidor gases
9. Llamar al centro de control (1860)
10. Poner el aire acondicionado a $\geq 29^{\circ}\text{C}$.

Valore los siguientes parámetros (1-muy mal/muy poco; 2- mal/poco; 3-regular; 4-bien/mucho; 5- muy bien/muchísimo)

| | 1 | 2 | 3 | 4 | 5 |
|--|---|---|---|---|---|
| ¿Cree que la formación le ha ayudado a adquirir conceptos que desconocía o no recordaba? | | | | | X |
| ¿Cree que la formación le ha ayudado a adquirir nuevas habilidades? | | | | X | |
| ¿Cree que la formación propicia la reflexión? | | | | | X |
| ¿Considera que la formación es interesante y fomenta la adquisición de nuevos conocimientos? | | | | | X |
| ¿Considera que el soporte digital es interesante y ayuda a adquirir nuevos conocimientos? | | | | | X |

Comentarios:

SIMULACIÓN OK. TIEMPO: 4:14.59

CUESTIONARIO DE LA PARTE TEÓRICA

Cuestionario relativo al procedimiento de puesta en marcha de las cabinas de salvamento ubicadas en las minas de potasa de Cataluña

Nombre: Andia Magaña Edad: 38 Sexo: F Fecha: 24/8/19

Escriba, en orden sucesivo de ejecución, todos los pasos que conozca para poner en marcha la cabina de salvamento

- 1- Terrar bien la preta.
- 2- Abrir las 2 bombonas de aire
- 3- Comprobar la posición de los botones del panel de control
- 4- Abrir las bombonas de oxígeno
- 5- Poner cinturones de calzada
- 6- Para cualquier problema llamar al telefono de centro de Control.

Valore los siguientes parámetros (1-muy mal/muy poco; 2- mal/poco; 3-regular; 4-bien/mucho; 5- muy bien/muchísimo)

| | 1 | 2 | 3 | 4 | 5 |
|--|---|---|---|---|---|
| ¿Cree que la formación le ha ayudado a adquirir conceptos que desconocía o no recordaba? | | | X | | |
| ¿Cree que la formación le ha ayudado a adquirir nuevas habilidades? | | | X | | |
| ¿Cree que la formación propicia la reflexión? | | X | | | |
| ¿Considera que la formación es interesante y fomenta la adquisición de nuevos conocimientos? | | X | | | |
| ¿Considera que el soporte digital es interesante y ayuda a adquirir nuevos conocimientos? | | | X | | |

Comentarios:

CUESTIONARIO DE LA PARTE PRÁCTICA

Cuestionario relativo al procedimiento de puesta en marcha de las cabinas de salvamento ubicadas en las minas de potasa de Cataluña

Nombre: Amelia Magaña Edad: 38 Sexo: F Fecha: 24/8/19

Escriba, en orden sucesivo de ejecución, todos los pasos que conozca para poner en marcha la cabina de salvamento

- 1- Cerrar bien la puerta
- 2- Abrir las 2 bombonas de aire
- 3- Mirar que los botones del panel estén en buena posición
- 4- Encender el funcionamiento automático
- 5- Pulsar el botón del motor de los cinturones
- 6- Cambiar de posición cuando lo requieran los cinturones
- 7- Abrir las bombonas de oxígeno que hacen falta
- 8- Llamar al teléfono de emergencia y dar mis coordenadas
- 9- Encender el medidor de gases
- 10- Encender el aire acondicionado.

Valore los siguientes parámetros (1-muy mal/muy poco; 2- mal/poco; 3-regular; 4-bien/mucho; 5-muy bien/muchísimo)

| | 1 | 2 | 3 | 4 | 5 |
|--|---|---|---|---|---|
| ¿Cree que la formación le ha ayudado a adquirir conceptos que desconocía o no recordaba? | | | | | X |
| ¿Cree que la formación le ha ayudado a adquirir nuevas habilidades? | | | | | X |
| ¿Cree que la formación propicia la reflexión? | | | | X | |
| ¿Considera que la formación es interesante y fomenta la adquisición de nuevos conocimientos? | | | | | X |
| ¿Considera que el soporte digital es interesante y ayuda a adquirir nuevos conocimientos? | | | | | X |

Comentarios: (SIMULACIÓN OK. TIEMPO: 5:36.10 min)

Mi aprendizaje sobre este procedimiento ha sido muy satisfactorio porque gracias a la visualización in situ de los hechos ocurridos en la cabina me ha ayudado a sobrellevar la situación correctamente.

CUESTIONARIO DE LA PARTE TEÓRICA

Cuestionario relativo al procedimiento de puesta en marcha de las cabinas de salvamento ubicadas en las minas de potasa de Cataluña

Nombre:

Ana

Edad:

60

Sexo:

F

Fecha:

24-8-2019

Escriba, en orden sucesivo de ejecución, todos los pasos que conozca para poner en marcha la cabina de salvamento

1º cerrar puerta

2º abrir el aire

3º controlar que las válvulas estén bien

4º poner los cartuchos

5º poner el oxígeno

6º encender el panel para que pase el aire por los cartuchos

7º

8º

9º Teléfono

10º aire acondicionado

Valore los siguientes parámetros: (1-muy mal/muy poco; 2- mal/poco; 3-regular; 4-bien/mucho; 5 muy bien/muchísimo)

| | 1 | 2 | 3 | 4 | 5 |
|--|---|---|---|---|---|
| ¿Cree que la formación le ha ayudado a adquirir conceptos que desconocía o no recordaba? | | | X | | |
| ¿Cree que la formación le ha ayudado a adquirir nuevas habilidades? | | X | | | |
| ¿Cree que la formación propicia la reflexión? | | X | | | |
| ¿Considera que la formación es interesante y fomenta la adquisición de nuevos conocimientos? | | | X | | |
| ¿Considera que el soporte digital es interesante y ayuda a adquirir nuevos conocimientos? | | | | X | |

Comentarios:

CUESTIONARIO DE LA PARTE PRÁCTICA

Cuestionario relativo al procedimiento de puesta en marcha de las cabinas de salvamento ubicadas en las minas de potasa de Cataluña

Nombre:

Ana

Edad:

60

Sexo:

F

Fecha:

24-8-2019

Escriba, en orden sucesivo de ejecución, todos los pasos que conozca para poner en marcha la cabina de salvamento

- 1º cerrar puerta
- 2º abrir botellas de aire
- 3º comprobar las valvulas
- 4º poner los cartuchos
- 5º poner en marche el absorbedor
- 6º poner el funcionamiento automatico
- 7º abrir botellas oxigeno
- 8º encender el medidor de gases
- 9º llamar por Telefono
- 10º poner el aire acondicionado

Valore los siguientes parámetros. (1-muy mal/muy poco; 2- mal/poco; 3-regular; 4-bien/mucho; 5-muy bien/muchísimo)

| | 1 | 2 | 3 | 4 | 5 |
|--|---|---|---|---|---|
| ¿Cree que la formación le ha ayudado a adquirir conceptos que desconocía o no recordaba? | | | | X | |
| ¿Cree que la formación le ha ayudado a adquirir nuevas habilidades? | | | | X | |
| ¿Cree que la formación propicia la reflexión? | | | | X | |
| ¿Considera que la formación es interesante y fomenta la adquisición de nuevos conocimientos? | | | | | X |
| ¿Considera que el soporte digital es interesante y ayuda a adquirir nuevos conocimientos? | | | | | X |

Comentarios: (SIMULACIÓN OK. TIEMPO: 15:50'34min)

ha sido una información muy interesante

CUESTIONARIO DE LA PARTE TEÓRICA

Cuestionario relativo al procedimiento de puesta en marcha de las cabinas de salvamento ubicadas en las minas de potasa de Cataluña

Nombre: Josep. Edad: 63 Sexo: M Fecha: 24-8-2019

Escriba, en orden sucesivo de ejecución, todos los pasos que conozca para poner en marcha la cabina de salvamento

- 1º/ cerrar la puerta
- 2º/ Abrir bombones de Aire
- 3º/ conectar el absorbedor.
- 4º/ comprobar que las válvulas conectadas conecta
- 5º/ colocar los cartuchos cal sodada. para mente
- 6º/ eliminar el CO₂
- 6º/ Comproban la tabla de presión según personas dentro
- 7º/ Abrir bombonas de oxígeno
- 8º/ comprobar niveles de CO₂
- 9º/ llevaron para que sepan que hay gente en la cabina.
- 10º/ Abrir aire acondicionado para regular la temperatura dentro de la cabina.

Valore los siguientes parámetros (1-muy mal/muy poco; 2- mal/poco; 3-regular; 4-bien/mucho; 5-muy bien/muchísimo)

| | 1 | 2 | 3 | 4 | 5 |
|--|---|---|---|---|---|
| ¿Cree que la formación le ha ayudado a adquirir conceptos que desconocía o no recordaba? | | | | X | |
| ¿Cree que la formación le ha ayudado a adquirir nuevas habilidades? | | | | X | |
| ¿Cree que la formación propicia la reflexión? | | | X | | |
| ¿Considera que la formación es interesante y fomenta la adquisición de nuevos conocimientos? | | | X | | |
| ¿Considera que el soporte digital es interesante y ayuda a adquirir nuevos conocimientos? | | | X | | |

Comentarios:

CUESTIONARIO DE LA PARTE PRÁCTICA

Cuestionario relativo al procedimiento de puesta en marcha de las cabinas de salvamento ubicadas en las minas de potasa de Cataluña

Nombre: Josép

Edad: 63

Sexo: M

Fecha: 24-8-2019

Escriba, en orden sucesivo de ejecución, todos los pasos que conozca para poner en marcha la cabina de salvamento

- 1º/ Cerrar Cabina
- 2º/ Abrir bombones de aire
- 3º/ Conectar Cabina.
- 4º/ Conectar absorbedor
- 5º/ Comprobar válvulas abiertas correctamente
- 6º/ Colocar botes de cal sodada
- 7º/ Comprobar concentración CO₂
- 8º/ Abrir bombones de oxígeno
- 9º/ Llamar que hay gente en la cabina.
- 10º/ Abrir aire acondicionado.

Valore los siguientes parámetros (1-muy mal/muy poco; 2- mal/poco; 3-regular; 4-bien/mucho; 5 muy bien/muchísimo)

| | 1 | 2 | 3 | 4 | 5 |
|--|---|---|---|---|---|
| ¿Cree que la formación le ha ayudado a adquirir conceptos que desconocía o no recordaba? | | | | | X |
| ¿Cree que la formación le ha ayudado a adquirir nuevas habilidades? | | | | X | |
| ¿Cree que la formación propicia la reflexión? | | | | X | |
| ¿Considera que la formación es interesante y fomenta la adquisición de nuevos conocimientos? | | | | X | |
| ¿Considera que el soporte digital es interesante y ayuda a adquirir nuevos conocimientos? | | | | | X |

Comentarios: (SIMULACIÓN OK. TIEMPO: 10:58'76 min)